

⑫ 公開特許公報(A) 平3-71275

① Int. Cl.³

G 06 F 15/66
G 09 G 5/36

識別記号

3 4 5

庁内整理番号

8419-5B
8839-5C

④ 公開 平成 3 年(1991) 3 月 27 日

審査請求 未請求 請求項の数 4 (全15頁)

④ 発明の名称 画像変換装置及び方法

⑥ 特 願 平1-206771

⑦ 出 願 平1(1989) 8 月11日

⑧ 発 明 者 中 川 哲 也 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑨ 発 明 者 馬 路 徹 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑩ 発 明 者 木 内 淳 東京都国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内

⑪ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地

⑫ 代 理 人 弁理士 小川 勝男 外1名

明 細 書

1. 発明の名称

画像変換装置及び方法

2. 特許請求の範囲

1. 所定の2次元離散データを座標変換して3次元グラフィックス用の2次元表示スクリーン上にマッピングすることを目的として、上記2次元離散データを座標変換して一度中間2次元離散データを生成し、続いて該中間2次元離散データを再び座標変換して上記2次元表示スクリーン上の画像を形成する2段階の処理過程を持つ画像変換装置において、上記2段階の処理過程の両者で入力離散データに対して補間フィルタリング処理及び間引きフィルタリング処理を適応的に行うことを特徴とする画像変換装置。
2. 特許請求の範囲第1項記載の画像変換装置において上記2段階の処理過程の両者で各々の座標変換によつて隣接する2つの出力離散データを入力データ上に逆写像した場合に逆写像された2データの間の隔が元の隣接間隔より狭くなる

領域の入力離散データに対しては補間フィルタリング処理を行い、広くなる領域の入力離散データに対しては間引きフィルタリング処理を行うことを特徴とする画像変換装置。

3. 所定の2次元離散データを座標変換して3次元グラフィックス用の2次元表示スクリーン上にマッピングすることを目的として、上記2次元離散データを座標変換して一度中間2次元離散データを生成し、続いて該中間2次元離散データを再び座標変換して上記2次元表示スクリーン上の画像を形成する2段階の処理過程を持つ画像変換方法において、上記2段階の処理過程の両者で入力離散データに対して補間フィルタリング処理及び間引きフィルタリング処理を適応的に行うことを特徴とする画像変換方法。
4. 特許請求の範囲第3項記載の画像変換方法において上記2段階の処理過程の両者で各々の座標変換によつて隣接する2つの出力離散データを入力データ上に逆写像した場合に逆写像された2データの間の隔が元の隣接間隔より狭くなる

領域の入力離散データに対しては補間フィルタリング処理を行い、広くなる領域の入力離散データに対しては間引きフィルタリング処理を行うことを特徴とする画像変換方法。

3. 発明の詳細な説明 〔産業上の利用分野〕

本発明は1次元以上の離散データに対して用いられるフィルタ装置に係り、特に座標変換に伴って発生する歪を除去する適応的なフィルタ方式に関するものである。

〔従来の技術〕

本発明に関係する公知例としては、公知例1: P.S. Heckbert, "Survey of Texture Mapping," IEEE CG&A, Nov. 1986, p.p. 56-67および公知例2: K.N. Pienta, "A Nonlinear, Real-Time Spatial Transform Technique," IEEE CG&A, Jan. 1986, p.p. 71-80を挙げることができる。

3次元グラフィックスで生成した3次元物体の表面は一般にあまりにも滑らかすぎるため、現実

感のない画像となってしまう。現実感を出すためにモデルを複雑にすると、モデル化のための労力や画像生成のための計算量が膨大なるものとなる。この問題を解決する代表的な方法がテクスチャ・マッピングと呼ばれる手法である。テクスチャ・マッピングは、3次元物体の表面にテクスチャと呼ばれる模様データをはりつけることによつて少ない労力と比較的少ない計算量で現実感を表現する手法である。テクスチャ・マッピングは座標変換とフィルタリングという2つの基本処理で構成されている。すなわち、まず第1図に示すように2次元平面(u, v)の格子点上の離散データとして定義されたテクスチャ画像データを座標変換し、3次元空間中の物体表面に張り付ける。次にこの3次元物体をさらに座標変換して2次元のスクリーン平面(x, y)上に投影する。第1図の例に示すようなスクリーン上で遠近感を持たせる変換は透視変換と呼ばれる最も基本的なものである。この変換はテクスチャ平面座標(u, v)とスクリーン平面座標(x, y)を用いて(1)式

のように定義される。ここでA, B, C, D, E, F, L, M, Nは変換係数であり、変換の種類によつて決定される。(1)式は2次元平面から2次元平面への変換であり、中間の3次元座標が消去されているため1次式の有理形となつていて非線形な変換となる。

$$\begin{aligned} u &= (Ax + By + C) / (Lx + My + N) \\ v &= (Dx + Ey + F) / (Lx + My + N) \end{aligned} \quad \dots (1)$$

(1)式に従つてテクスチャ平面上の各格子点で定義された画像データを座標変換した場合、第1図のテクスチャ平面上の格子点Pは必ずしも、スクリーン上の格子点に変換されるとは限らない。第1図に示すようにスクリーン上ではPは格子点の間に写像される可能性が大きい。そこで、スクリーン上の格子点における画像データを求めるためにその格子点の近くに写像されてくる複数のテクスチャ画像データに適切な重みを付けて足しあわせることが必要となる。この重みづけ演算のことをフィルタリング、この重み係数のことをフィルタ係数と呼ぶ。従来このフィルタリングは

第1図に示す手順で行われていた。ここではスクリーン平面上の格子点Lに注目し、この格子点の画像データを求めることを考える。まず、Lの近傍を定義するためにL1の勢力範囲PBを考える。第1図ではL1の勢力範囲PBは隣接の格子点の勢力範囲と多少オーバーラップする円で定義されているが、この決め方は恣意的であつてオーバーラップしない正方形で定義する場合もある。次に、この円PBを式(1)に従つてテクスチャ平面上に逆変換する。この変換で円PBは楕円PB'に変形される。楕円PB'の内側にあるテクスチャ平面上の格子点T1, T2, T3, T4を式(1)に従つて変換しスクリーン平面上の座標を求める。この場合、これら4つの格子点T1, T2, T3, T4が格子点L1の画像データを求めるために使用される近傍データとなる。一方、スクリーン平面上では、格子点L1を中心とし、ここからの相対座標の関数としてフィルタ係数が定義されている。第1図に低解度から高解度までの5種のフィルタ形状が

示されている。第12図の例では比較的単純なトライアングル型が用いられている。第12図ではフィルタの定義域が円P Bであり、かつフィルタ係数の大きさを高さで示してあるため、フィルタ係数全体が円P Bを底面とする円錐として表されている。すなわち、この円錐の断面が第13図に示されているトライアングル型となつている。よつてスクリーン平面上に写像されたT1, T2, T3, T4のそれぞれの座標位置におけるこの円錐の高さにそれぞれの画像データをかけて加え合わせるにより、格子点L1の画像データを求めることができる。しかしながら、この方法ではテキスト画の周辺部分のように(1)式の変換で大きく変形する部分での演算量増加が問題となる。例えば、1000画素×1000画素のテキスト画像を逆視変換する場合、テキスト画周辺の1000画素がスクリーン上で数百画素に対応する場合がある。この場合、スクリーン上の1格子点の計算のために数百ものテキスト・データを座標変換する必要が生じてしまい、これに伴う座

標変換の演算量の増加が問題となつていた。

そこで公知例1では補間に必要な多量のテキスト・データをスクリーン上に写像する代わりに比較的データ量の少ないフィルタの係数をテキスト平面に逆写像し、テキスト平面でフィルタリングすることにより座標変換に伴う演算量を低減する方法が提案されている。この方法を第12図の例に適用するとスクリーン上の1つの格子点に対して必要な座標演算は以下のようになる。すなわち、この格子点自身をテキスト平面上に写像する座標変換1回とスクリーン平面の円P B上で定義されているフィルタをテキスト平面上に写像するのに必要な5つのパラメータ(楕円P B'を決定する中心位置の指定に2つ、半径の指定に2つ、回転角の指定に1つの合計5つ)の計算である。この5つのパラメータの計算量はおおよそ座標変換2回分の計算量に等しい。このため、スクリーン上の全ての格子点で一律、座標変換3回分程度の計算量となり、1格子点当たりの座標演算量を大幅に減少できる。しかし、楕円P B'は対

応するスクリーン平面上の格子点ごとに全て異なるので画面全体ではまだまだ膨大な演算量を必要とする。例えば、1000画素×1000画素の画面を扱う場合、スクリーン上の格子点をテキスト平面上に写像する座標変換1,000,000回と楕円P B'を決定するのに必要なパラメータ演算5,000,000回を必要とする。そこで、さらに演算量は減らすことが課題となつている。また、公知例1はサンプリング定理に基づいた高精度の補間を考える場合の問題点を含んでいる。すなわち、第13図に示すようにサンプリング定理に基づく理想精度のフィルタ形状は無限の定義域を必要とするにもかかわらず、公知例1では用いられるフィルタの定義域は第12図に示すように断片的であるものの予め固定的に決められていた。このため、一度ある精度のフィルタリングを実行するシステムを公知例1を用いて実現するとフィルタの定義域が固定されてしまう。よつてフィルタ係数を入れ替えられてもフィルタの定義域を広げることができず、フィルタを理想精度フィルタによ

り近づけるという意味でより高精度のフィルタリングを実現する場合の障害となつていた。しかし、だからといってフィルタの定義域を予め充分広くとつておくと、低精度なフィルタリングで充分な場合に無駄な演算量が増加してしまうという問題点があつた。又、1格子点の計算ごとにテキスト平面上で異なる形状の楕円領域内データを用いるため、これに伴つて煩雑なメモリ・アクセスが必要となる問題点があつた。

フィルタリング処理に伴う座標演算量を低減するもう1つの方法として公知例2では座標変換に2パスアルゴリズムを利用したものが提案されている。2パスアルゴリズムは第8図に示したように2次元の座標変換を縦軸、横軸2回の1次元の座標変換に分解する方法である。第8図ではスクリーン平面(x, y)とテキスト平面(u, v)のほかに中間画像平面(x, v)を考える。スクリーン平面(x, y)とテキスト平面(u, v)は式(1)で関係付けられていた。スクリーン平面(x, y)と中間画像平面(x, v)はxを固定

し、以下に示す式(2) (式(1)の一方)を用いて関係づけられる。すなわち、スクリーン平面(x, y)の縦一列(第x列)は中間画像平面(x, v)

$$v = (Dx + Ey + F) / (Lx + My + N) \quad \cdots(2)$$

の縦一列(第x列)とのみ対応するので処理は1次元となる。中間画像平面(x, v)とテキストヤ平面(u, v)はvを固定し、以下に示す式(3) (式(1)のもう一方)を用いて関係づけられる。ただし、式(3)の右辺に差数yが含まれているので、このyは式(2)をyについて解いた $y = f(x, v)$ で置き換えて消去する必要がある。

$$u = (Ax + By + C) / (Lx + My + N) \quad \cdots(3)$$

この時、vが固定されているので中間画像平面(x, v)の横一行(第v行)はテキストヤ平面(u, v)の横一行(第v行)とのみ対応するので処理は1次元となる。第8図の変換式を整理した結果を第9図に示す。これによると、2パスの各パスにおける1次元の座標変換は変換係数が異なるだけで同じ形、すなわち1変数1次式の有理形をしていることがわかる。(例えば、 $u = (0.5$

$+ d) / (a \cdot x + b)$)。このアルゴリズムは1次元中間画像を生成するため、画像データへのアクセス回数が2倍になる。しかし、補間処理のためのデータアクセスを2次元でなく、1次元として扱えるという特長がある。公知例2はこの特長を生かしてフィルタという概念を用いないで実際に下記に説明する低精度の補間を行う直線的方法を提案している。この方法の詳細を第14図を用いて説明する。第14図(a)は1次元の画面入力データI1, I2, I3, I4を1.33倍に拡大して画面数を増加させるとともに画像情報の横情報を含む画面出力データO1, O2, O3, O4, O5を求める例を示している。ここで入力データI1~I4はテキストヤ画像のu座標に相当し、出力データO1~O5は中間画像のx座標に相当する。第14図(b)は1次元の画面入力データI1, I2, I3, I4, I5を0.75倍に縮小して逆に画面数を減少させるとともに画像情報の横情報を含む画面出力データO1, O2, O3, O4を求める例を示している。第

14図(a)では入力を1.33倍するということは出力1個を得るのに入力の0.75個分(1/1.33)あればよいという考え方を using している。まず、出力O1は入力I1の0.75個分を用いて得られる。この時、入力I1の0.25個分は使われずにまだ残っている。次に、出力O2は入力I1のこの0.25個分と入力I2の0.5個分の合計0.75個分を用いた横和演算によって得られる。すなわち、 $(0.25 \times I1 + 0.5 \times I2) / 0.75$ を計算すればよい。ここで0.75で割るのは正規化のためである。この時点で入力I2の0.5個分はまだ使われずに残っている。出力O3はこの残っている入力I2の0.5個分と入力I3の0.25個分を用いて得られる。以下、同様に求められる。第14図(b)では入力を0.75倍ということは出力1個を得るのに入力の1.33個分(1/0.75)あればよいという考え方を using している。まず、出力O1は入力I1の1個分と入力I2の0.33個分の合計1.33個分を用いて得られる。すなわち、

$(1 \times I1 + 0.33 \times I2) / 1.33$ を計算すればよい。この時点で入力I2の0.67個分はまだ使われずに残っている。次に出力O2はこの残っている入力I2の0.67個分と入力I3の0.66個分を合計した入力の1.33個分を用いて得られる。以下、同様に求められる。このように、この方法にはフィルタという概念がないためフィルタリング処理に伴って必要となる座標変換を全く必要とせず非常に高速の処理が可能となる。しかしながら、逆にフィルタという概念が無いため、公知例1のように第13図のより高精度のフィルタ係数を用いることによって、より高精度な補間を行うということが不可能という問題点があった。又、この方法は第14図に示すように1次元データを左から右に向かって処理する場合、各画面ごとで発生する座標の狂いを右側に次々としお寄せするため画像の右端で座標が大きく狂ってしまうという本質的な問題点を持っている。この座標の狂いによって画像全体としては周辺部分にノイズが発生するという問題が生じる。

(発明が解決しようとする課題)

本発明は本発明者による上記の検討結果を基にしてなされたものであり、その目的は下記の技術課題を解決することにある。

- (1) 公知例1のフィルタリング処理で用いられるフィルタの定義域は恣意的であるもの予め固定的に決められていた。このため、一度ある程度のフィルタリングを実行するシステムを実現するとフィルタの定義域が固定されてしまう。よって高精度のフィルタ係数はより広い定義域を必要とするのにフィルタの定義域を広げることができず、フィルタを粗精度フィルタにより近づけるという意味でより高精度のフィルタリングを実現する場合の障害となっていた。しかし、だからといってフィルタの定義域を予め充分広くとっておく、低精度なフィルタリングで充分な場合に無駄な演算量が増加してしまうという問題点があった。一方、公知例2にはフィルタという概念が無いため、公知例1のようにより高精度のフィルタ係数を用いることに

よって、より高精度な補間を行うということが本質的に不可能という問題点があった。従って本発明の第1の目的はフィルタリングシステムを実現した後でもフィルタの定義域が固定されずフィルタ係数を入れ替えることによりサンプリング定理に基づく理想的な精度にいくらかでも近づけるフィルタリングを可能とする手段を提供することである。

- (2) 公知例1ではスクリーン画像のフィルタリングに伴う座標演算処理を低減するため、データ量の多いテキストチャネルデータをスクリーン平面上に写像する代りにデータ量の少ないフィルタ係数をテキストチャネルに写像していた。しかし、それでも画面全体でまだまだ膨大な座標演算量を必要とし、さらにこの演算量を減らすことが課題となっていた。一方、公知例2ではフィルタを使用しない直感的な方法でこのフィルタリングに伴う座標演算量を全く省略して高速処理を実現していた。しかし、この方法では変換後の座標位置が不正確になり結果の画像にノイ

ズが生じてしまうという本質的な問題点を持っていた。従って本発明の第2の目的は正確な座標変換を行ないながら座標演算量を最小にするフィルタリングの手段を提供することにある。

- (3) 公知例1ではフィルタリングのためにスクリーン上の1格子点の計算ごとにテキストチャネル上で異なる形状の楕円領域内データを用いるため、これに伴って煩雑なメモリ・アクセスが必要となる問題点があった。従って本発明の第3の目的は単純なメモリ・アクセスを用いて実現できるフィルタリングの手段を提供することにある。

(課題を解決するための手段)

上記の種々の目的を解決するために本発明は下記の特徴を有する。

- (1) 第1、第2の目的は従来スクリーン平面上で定義されていたフィルタをテキストチャネル上で定義することにより達成される。
- (2) 第3の目的は座標変換の2パス・アルゴリズムを用いて各パスごとに1回ずつ合計2回の1

次元フィルタリングを行なうことにより達成される。

(作用)

- (1) 第1の解決手段によるとフィルタ係数がテキストチャネル上で定義されることになる。これにより、公知例1ではスクリーン平面上で定義されたフィルタ係数をテキストチャネルに写像するために必要だったスクリーン平面上のフィルタ定義域が不要となる。よって、使用したい精度のフィルタ係数に合わせて、テキストチャネル上でその定義域を自由に決定できるので上記第1の目的を達成できる。又、フィルタ係数がフィルタリング処理に用いるテキストチャネルデータと同一のテキストチャネル上で定義されているため、公知例1のようにフィルタ係数をテキストチャネルに写像する処理自体が不要になり上記第2の目的を達成できる。
- (2) 第2の解決手段である座標変換の2パス・アルゴリズムを用いると、1回の2次元のフィルタリングを2回の1次元のフィルタリングで置

き換えることができる。このため、メモリアクセスは単純な1次元アクセスとなるので上記第3の目的を達成できる。また、このように本発明は基本的に1次元処理に基づいているので画像以外の任意の1次元データに適用可能である。

(実施例)

まず、実施例を説明するに先立つて、本発明の原理的背景を述べる。前述したように本発明は第9図の座標変換の2パス・アルゴリズムに基づいている。このため、第9図に示すように前半パスで中間画像平面上の1次元格子点データをテキストチャ平面上の1次元格子点データから求め、次に後半パスでスクリーン平面上の1次元格子点データを中間画像平面上の1次元格子点データから求めるという手順になる。この手順を前半パスについて、より詳細に説明する。まずテキストチャ平面上の離散的な1次元格子点データからテキストチャ平面上の1次元連続データを求める。続いて第10図に図示する1次元座標変換式に従って中間

画像平面上の1次元格子点座標をテキストチャ平面上に写像する。第10図中で x_o-1 、 x_o 、 x_o+1 は中間画像平面上の1次元格子点座標であり、 $u(x_o-1)$ 、 $u(x_o)$ 、 $u(x_o+1)$ はこれらの点がテキストチャ平面上に写像された時の1次元座標である。 x_o と $u(x_o)$ の関係は下の式(4)で与えられる。

$$u(x_o) = (ox + d) / (ax + b) \quad \cdots (4)$$

ここで、係数 a 、 b 、 o 、 d は第9図中のものと同じである。

そして最後にこの写像された点 $u(x_o-1)$ 、 $u(x_o)$ 、 $u(x_o+1)$ でテキストチャ平面上の1次元連続データを再サンプリングすればよいわけである。すなわち、第10図の中間画像の等間隔の格子点 x_o-1 、 x_o 、 x_o+1 の画像データを求めるために $u(x) = (ox + d) / (ax + b)$ の変換式に従ってテキストチャ画像上の座標データを計算する必要がある。しかし、この変換式に従って格子点 x_o-1 、 x_o 、 x_o+1 をテキストチャ画像に投影すると非等間隔の座標位置の画像データが

必須となる。従って、この非等間隔の座標位置の画像データを求めるために、等間隔の格子点で与えられる画像データからサンプリングする前の連続データを再生し、これを非等間隔でサンプリングすれば良い。しかしながらこの時、中間画像平面上からテキストチャ平面上に写像された隣接する2つの格子点の間隔が格子点座標によってテキストチャデータの格子点間隔より狭くなったり、広くなったりする。例えば、第10図中で $u(x_o-1)$ と $u(x_o)$ の間隔はテキストチャデータの格子点間隔より狭くなっている。このように再サンプリングの間隔が座標位置によって変化するので各座標位置ごとにサンプリング定理を考慮し、折り返し歪が起らないように注意してフィルタリングする必要がある。すなわち、第10図の開引きの部分ではテキストチャ画像の4つの画素が中間画像の2つの画素に開引かれ、逆に補間の部分ではテキストチャ画像の2つの画素から中間画像の4つの画素が補間されているので折り返し歪が起らないように、この2つの部分でそれぞれ適当なフ

ィルタリングが必要である。

次に、この議論の理解を容易にするために第15図～第17図を用いて折り返し歪とサンプリング定理について詳細に説明する。第15図(a)は1次元の連続データと、それをフーリエ変換して得られた角周波数スペクトル特性である。この連続データが含む最大角周波数は ω であり、それ以上の成分は含まない。以下、このように角周波数成分が帯域制限された連続データのみを対象と考える。第15図(b)は第15図(a)の連続データを間隔 T_o でサンプリングして離散化した場合の角周波数スペクトルは元の角周波数スペクトルが $2\pi/T_o$ の周期を持つて配置されたものになることを示している。第15図(c)はサンプリング間隔 T_o が大き過ぎて $\pi/T_o < \omega$ となる場合である。この場合は図に示すように斜線部分の角周波数スペクトルが重なることにより折り返し歪と呼ばれる歪が生じる。第16図(e)に示すように、この重なりが生じないような充分に小さな間隔 $T_o(\pi/T_o \geq \omega)$ でサンプリングされ

たデータにカットオフ周波数 π/T_0 のローパスフィルタをかけて角周波数領域で破綻に置まれる部分だけ取り出すと第16図(b)に示すように連続データを再生できる。信号処理の技術分野ではこのことをサンプリング定理と呼んでいる。よつて第16図(a)に示すようにサンプリング定理を満たす間隔 T_0 でサンプリングされたデータをそれより小さな間隔 T_x で再サンプリングして x_0 におけるデータを求めるためには、まず、第16図(a)の離散データにカットオフ周波数 π/T_0 のローパスフィルタをかけて第16図(b)に示す連続データに補間し、続けて第16図(c)に示すようにこの連続データを間隔 T_x で再サンプリングして x_0 におけるデータを求めればよい。この再サンプリングの過程で連続データは角周波数が π 以上の成分を含まず、また、 $T_0 > T_x$ より $\pi/T_x > \pi/T_0$ となつていのでサンプリング定理を満たしており折り返し歪は生じない。

しかし、もし第16図(c)における再サン

プリングの間隔 T_x が T_0 より大きい場合($T_x > T_0$)サンプリング定理を満たしておらず、第15図(c)と同様に折り返し歪が発生してしまう。そこで、この場合は第17図(a)に示すように間隔 T_0 でサンプリングされた離散データにカットオフ周波数 π/T_x のローパスフィルタをかけて角周波数領域で破綻に置まれる部分だけ取り出して角周波数が π/T_x より大きな成分を持たないように帯域制限された連続データに補間する。この様子を第17図(b)に示す。そして続けて第17図(c)に示すように間隔 T_x で再サンプリングして x_0 におけるデータを求めればよい。このようにすると再サンプリングの過程で連続データは角周波数が π/T_x 以上の成分を含まないのでサンプリング定理を満たしており折り返し歪は生じない。

以上述べてきたことをまとめると再サンプリングの間隔 T_x がテキスチャデータの間隔 T_0 より小さい場合は連続的なテキスチャデータを再生するためには離散テキスチャデータにカットオフ

周波数が π/T_0 に固定されたローパスフィルタをかければよい。また、再サンプリングの間隔 T_x がテキスチャデータの間隔 T_0 より大きい場合には折り返しのない連続的なテキスチャデータを再生するためにサンプリング定理を満たすように離散テキスチャデータにカットオフ周波数 π/T_x のローパスフィルタをかける必要がある。ただし、ここで T_x は逆写像された2つの格子点の間隔なので可変である。従つて第10図の関引きの部分ではカットオフ周波数 π/T_x の可変なローパスフィルタを用いる。

そこで、本発明においてはカットオフ周波数を π/T_0 に固定した補間用ローパスフィルタとカットオフ周波数 π/T_x が T_x によつて可変となる関引き用ローパスフィルタからなる適応フィルタリング手段及びこれら2つのフィルタのうちどちらか一方を選択する制御を行うフィルタ制御手段を備えた。これにより、第9図の2パスのうちの前半パスにおいては第10図に示すように中間画像平面上の各格子点 x_0-1, x_0, x_0+1

をテキスチャ平面上に逆写像し、逆写像された格子点間隔 T_x がテキスチャデータ間隔 T_0 より狭い場所(補間の部分)の格子点データはカットオフ周波数を π/T_x に固定した補間用ローパスフィルタをテキスチャデータにかけ求めることができる。逆写像された格子点間隔 T_x がテキスチャデータ間隔 T_0 より広い場所(関引きの部分)の格子点データはカットオフ周波数 π/T_x が T_x によつて可変となる関引き用ローパスフィルタをテキスチャデータに適応的にかけて求めることができる。後半パスにおいても同様に、各格子点ごとにサンプリング定理を満たすように、中間画像を適応的にフィルタリングしてスクリーン画像を求めることができる。

以下、本発明の第1の実施例を第1図から第4図までを用いて説明する。本実施例は本発明を専用ハードウェアを用いて実現した例である。第1図は第1の実施例の基本構成を示している。第1図は遠近感換の2パス・アルゴリズムを実現するための構成である。第1図は、2つの適応フィル

タ(5)、(5')、2つのフィルタ制御装置(7)、(7')から構成される。また、説明のために、本発明の処理の対象となる画像データ(テキスト画、中間画、スクリーン画)も、合わせて示してある。適応フィルタ(5)とフィルタ制御装置(7)は第9図の前半の1パスを処理する部分であり、テキスト画を入力して中間画を出力する。適応フィルタ(5')とフィルタ制御装置(7')は後半の1パスを処理する部分であり、中間画を入力してスクリーン画を出力する。前半の1パス処理と後半の1パス処理は入力画像のアクセスが行単位か列単位かが異なるだけで後はすべて同一であるので、ここでは前半の1パス分の説明だけを行なう。フィルタ制御装置(7)は逐次演算のための座標変換係数(6)と中間画の一面分の座標データ(2)を入力して適応フィルタの中心位置データ(3)とサンプル間隔データ(4)を生成する。適応フィルタ(5)はフィルタ制御装置(7)から適応フィルタの中心位置データ(3)とサンプル間隔

データ(4)を入力してテキスト画の一行分のデータ(1)を用いて中間画の一面分のデータ(1')を生成する。生成する画像データの座標はフィルタ制御装置(7)に入力された値(2)となる。中間画の座標データを左上の原点から右下に向かつて方向に全画送スキャンすることによって1パス分の処理が完了する。

次に、第1図のフィルタ制御装置の処理の詳細を第2図を用いて説明する。第2図はフィルタ制御装置の全体を示す図である。第2図のフィルタ制御装置は座標計算装置(8)、差の絶対値演算装置(9)、2つの遅延装置(10、10')、最大値出力装置(11)から構成されている。座標計算装置(8)は座標変換係数 a, b, c, d (6)と中間画の一面分の座標値 x からテキスト画における座標値 u を計算する。ここで座標変換係数 a, b, c, d は第9図中のものと同じであり、予め計算されているものとする。よって座標計算装置(8)は乗算2回、加算2回、除算1回を計算する。ハードウェアによる実現法は

単純に乗算器2個、加算器2個、除算器1個で構成できる。しかし、ハード量を減らすために加算器1個で構成し、これをマイクロプログラムなどで多量に使用して乗算や除算を実現することもできる。又、ここで求めた u は適応フィルタの中心位置を示すデータとなる。第2図では説明の都合上、中間画の座標値 $x+1$ からこれに対応するテキスト画の座標値 $u(x+1)$ を求める例が示してある。2つの遅延装置(10、10')は座標データ x の一面分を遅延を行うので遅延装置(10')の出力は $u(x_0)$ (3)となる。差の絶対値演算装置(9)は座標計算装置(8)の出力である座標値 $u(x+1)$ と遅延装置(10')の出力 $u(x_0)$ を入力してこれら入力値の差の絶対値を演算する。最大値出力装置(11)は差の絶対値演算装置(9)の出力である差の絶対値 $|u(x+1)-u(x_0)|$ と遅延装置(10)の出力 $|u(x_0)-u(x_0-1)|$ を入力してどちらか大きい方をサンプル間隔データ $T(x_0)$ (4)として出力する。すなわち、この $T(x_0)$ (4)は中間画

上の座標値 x_0 がテキスト画平面上に逆写像された時の座標値 $u(x_0)$ とその前側に逆写像されている $u(x_0+1)$ 、 $u(x_0-1)$ との距離のうちの大きい方となる。又、この時 $u(x_0)$ はテキスト画平面上の格子点にあるとは限らない。 $u(x_0)$ の画像データを求めるためには、まずテキスト画平面上の格子点で定義されている離散的なテキスト画データの間を連続的に補間し、次にこの補間された連続データを間隔 $T(x_0)$ で再サンプリングすればよい。しかし、 $T(x_0)$ がテキスト画・データ間隔 T_0 より大きい場合、サンプリング定理を満足するようにテキスト画データの横断方向の空間周波数が $\pi/T(x_0)$ より大きな成分を持たないようにローパスフィルタをかける必要がある。次に、第1図の適応フィルタの処理の詳細を第3図、第4図を用いて説明する。第3図は適応フィルタの全体を示す図である。適応フィルタ(5)は2つのローパス・フィルタ(12)、(13)とセレクタ(14)から構成される。ローパス・フィルタ(12)はカutoff周波数

f_0 が π/T_0 に固定されている。ローパス・フィルタ (13) はカットオフ周波数 f_0 が π/T_0 であり、適応フィルタの入力 Tx_0 (3) によつて可変となる。セレクト (14) は Tx_0 (3) と原画のサンプリング間隔 T_0 との大小によつて一行分の入力画像 (1) に2つのローパス・フィルタ (12), (13) のどちらをかけて出力となる中間画像の1画素 (1') を生成するかを選択する。適応フィルタの入力 $u(x_0)$ (4) は2つのローパス・フィルタ (12), (13) の両者に入力されており、一行分の入力画像 (1) 上のフィルタの中心位置を決定する。第4図にこの適応フィルタのカットオフ周波数 f_0 の入力 Tx_0 (3) 依存性を示す。第4図の縦軸は適応フィルタの入力 Tx_0 (3)、縦軸は適応フィルタのカットオフ周波数 f_0 である。カットオフ周波数 f_0 は $Tx_0 \leq T_0$ の時、一定値 π/T_0 となり、 $T_0 < Tx_0$ の時、 $f_0 = \pi/T_0$ と可変になる。新してサンプリング間隔 Tx_0 が原画のサンプリング間隔 T_0 より小さい場合に座標 $u(x_0)$ にお

ける画像データを求めるには、間隔 T_0 で並んでいる一行分のテキストチャデータの格子点間のデータを単に連続的に補間し、続いて間隔 Tx_0 で再サンプリングすれば良い。なぜなら、この時は $\pi/Tx_0 \geq \pi/T_0$ であり、テキストチャデータの横軸方向の空間周波数は、もともと π/T_0 より大きな成分を持っていない。このため連続的に補間されたテキストチャデータを間隔 Tx_0 で再サンプリングしてもサンプリング定理を満たしており、折り返し歪は発生しない。すなわち、 $Tx_0 \leq T_0$ の時はテキストチャデータの格子点間のデータを補間するためにカットオフ周波数 $f_0 = \pi/T_0$ のローパスフィルタをかければ良い。第4図の $Tx_0 \leq T_0$ の場合がこれに対応している。またサンプリング間隔 Tx_0 が原画のサンプリング間隔 T_0 より大きい場合に座標 $u(x_0)$ における画像データを求めるには、間隔 T_0 で並んでいる一行分のテキストチャデータの格子点間のデータを連続的に補間した後、間隔 Tx_0 で再サンプリングする前にカットオフ周波数 $f_0 = \pi/Tx_0$ のローパス

フィルタをかける必要がある。なぜなら、この時は $\pi/Tx_0 \leq \pi/T_0$ でありテキストチャデータの横軸方向の空間周波数が π/Tx_0 より大きな成分を持っている可能性があるからである。結局この場合はテキストチャデータにカットオフ周波数 $f_0 = \pi/T_0$ のローパスフィルタをかけて連続値補間した後、さらに周波数 $f_0 = \pi/Tx_0$ のローパスフィルタをかけることになるが $\pi/Tx_0 \leq \pi/T_0$ なので後者の処理のみ行えば良いことがわかる。第4図の $Tx_0 \geq T_0$ の場合がこれに対応している。

次に第5図から第7図を用いて本発明の第2の実施例を説明する。本実施例はDSP (デジタル・シグナル・プロセッサ) やマイクロ・プロセッサのようにプログラム可能な汎用のハードウェアを用いて本発明を実現した例である。第5図は第2の実施例の基本構成を示しており、逆変換の2パス・アルゴリズムを実現するための構成である。第5図はプログラム可能なDSP又は、マイクロ・プロセッサのような類似装置 (15)、

(16) で構成される。また、説明のために、本発明の処理の対象となる画像データ (テキストチャ画像、中間画像、スクリーン画像) も、合わせて示してある。プログラム可能なDSP (15) は前半の1パスを処理するプログラムを命令メモリ中に格納しており、テキストチャ画像を入力して処理し、中間画像を出力する。プログラム可能なDSP (16) は後半の1パスを処理するプログラムを命令メモリ中に格納しており、中間画像を入力して処理し、スクリーン画像を出力する。第5図ではプログラム可能なDSP (15), (16) が2つ存在するがプログラム可能なDSPを1つだけ用意し、命令メモリの内容を入れ替えて多重使用するようにもできる。第6図に前半の1パスを処理するプログラムのフローを第7図に後半の1パスを処理するプログラムのフローを示す。前半の1パス処理と後半の1パス処理は入力画像のアクセスが行単位が列単位が異なるだけで後はすべて同一であるので、ここでは第6図を用いて前半1パス分のプログラム・フローの説明だけを

行なう。まず、中間画像の全ての行（第 v 行）について以下の2つの処理を繰り返す。すなわち、まず第1に第6図中の式に従って座標変換係数 a 、 b 、 c 、 d を計算する。第2に、これを用いて各行ごとに適応フィルタリング演算を行なう。ここで、各行ごとの適応フィルタリング演算では以下の3つの処理を全ての列 x について繰り返す。まず、第1に座標変換係数 a 、 b 、 c 、 d と列座標 $x-1$ 、 x 、 $x+1$ を用いて $u(x-1)$ 、 $u(x)$ 、 $u(x+1)$ を計算する。第2に $|u(x)-u(x-1)|$ 、 $|u(x+1)-u(x)|$ を計算し、大きい方を Tx とする。第3に Tx と T_0 （テキストヤ平面上の格子点間隔）との大小関係により以下の処理を行なう。 $Tx > T_0$ の場合はカットオフ周波数 $f_0 = \pi / Tx$ の1次元ローパスフィルタをテキストヤ画像の第 v 行にかけて中間画像の第 v 行、第 x 列の格子点の値を求める。 $Tx < T_0$ の場合はカットオフ周波数 $f_0 = \pi / T_0$ の1次元ローパスフィルタをテキストヤ画像の第 v 行にかけて中間画像の第 v 行、第 x 列の格子点の値を求

める。以上のプログラムフローによつて第1の実施例では専用ハードウェアで行なっていたのと同じ前半の1パス分の処理を実現できる。又、第7図のプログラムフローによつて第1の実施例では専用ハードウェアで行なっていたのと同じ後半の1パス分の処理を実現できる。

このように本発明を用いられれば中間画像上の各格子点データを求める処理を1次元のテキストヤデータ上に逆写像された中間画像上の各格子点における格子点間隔を用いてテキストヤデータを再サンプリングすることによつて実行できる。このため、中間画像上の各格子点ごとにサンプリング定理に基づいた折り返し歪を考慮することが可能になる。そこで、補間や間引きに用いるローパスフィルタとして振幅がカットオフ周波数 f_0 以下で1、 f_0 以下で0となる理想的なものを用いられれば理論的に歪を全く含まない補間処理や間引き処理を考慮することが可能となる。よつてローパスフィルタとして、より理想に近いものを用いることによつてこの理想的な場合に近づけるという意味で

実際にいくらかでも高精度な補間や間引きが可能となる。又、これらのローパスフィルタは処理を行う1次元のテキストヤデータと同一平面上で定義されているので、もしもローパスフィルタがスクリーン平面上や中間画像平面上で定義されていた時に必要となる余分な座標変換は不要となる。以上の説明は座標変換の2パスの内の前半についてであるが、後半パスについても全く同様のことがいえる。又、用いられるフィルタ処理は全て1次元でのメモリアクセス単純な1次元アクセスとなる。

（発明の効果）

以上、説明してきたように本発明のフィルタ装置は2次元スクリーン上の各格子点における画像データを求める処理をテキストヤ平面及び中間画像平面上の1次元データに対する適応的な1次元の補間処理、間引き処理として実行することができるためサンプリング定理に基づく厳密な考察が容易となる。又、補間処理、間引き処理用の1次元フィルタを理想的なローパスフィルタに近づけ

ることによつて、いくらかでも高精度な補間処理、間引き処理が可能となるという大きな効果がある。さらに補間処理、間引き処理用のフィルタをデータと同一の平面上で定義しているのでフィルタリングに伴うフィルタ係数やデータの座標変換を完全に避けることができるという大きな効果がある。又、用いられるフィルタ処理は全て1次元なのでメモリアクセスも単純な1次元となり、複雑なメモリアドレスリングが避けられるという大きな効果がある。

4. 図面の簡単な説明

第1図は本発明の第1実施例の基本構成図、第2図は第1図中で使用されているフィルタ別構築の内部構成図、第3図は第1図中で使用されている適応フィルタの内部構成図、第4図は第3図の適応フィルタのカットオフ周波数の再サンプリング間隔 T 依存性を示す図、第5図は本発明の第2実施例の基本構成図、第6図および第7図は本発明をプログラム可能な汎用ハードウェアで実現する場合に用いられるプログラムのフロー図、第

ム可能なDSP又は類似装置。

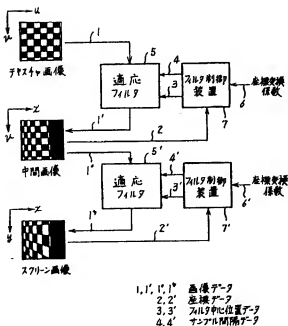
代理人 井理士 小川 勝



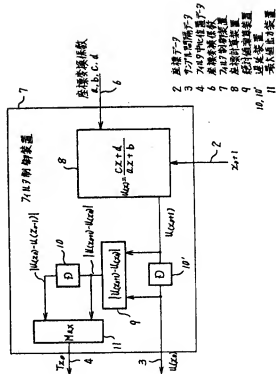
8面および第9図は2次元画像の座標変換の2パスアルゴリズムを説明する図、第10図は第9図の前半パスを詳細に説明した図、第11図は3次元グラフィックスで用いられる代数的な座標変換である透視変換を直観的に示した図、第12図は公知例1のアルゴリズムを説明する図、第13図は本発明及び公知例で用いられる1次元フィルタの形状を示す図、第14図は公知例2のアルゴリズムを説明する図、第15図は折り返し延とサンプリング定理を説明した図、第16図および第17図はサンプリング定理と本発明の理論的關係を説明した図。

1, 1', 1'', 1''' …画像データ、2, 2' …座標データ、3, 3' …フィルタ中心位置データ、4, 4' …サンプル間隔データ、5, 5' …適応フィルタ、6, 6' …座標変換係数、7, 7' …フィルタ制御装置、8 …座標計算装置、9 …絶対値演算装置、10, 10' …遅延装置、11 …最大値出力装置、12, 13 …ローパスフィルタ、14, 14' …セクタ、15, 16 …プログラ

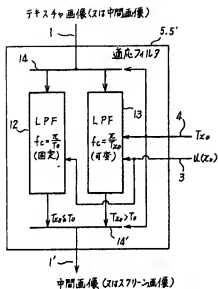
第 1 図



第 2 図

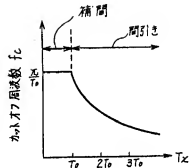


第 3 図

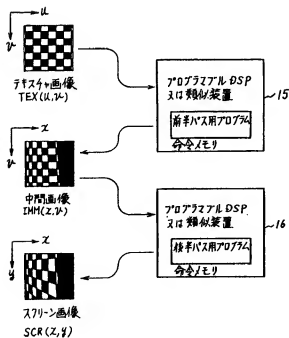


12, 13 ローパス・フィルタ
14, 14' セレクタ

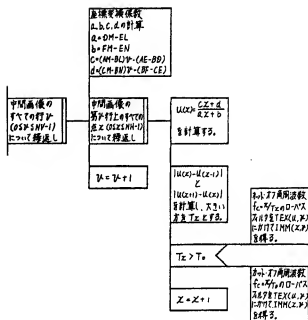
第 4 図



第 5 図

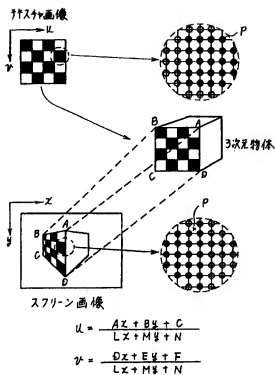


第 6 図

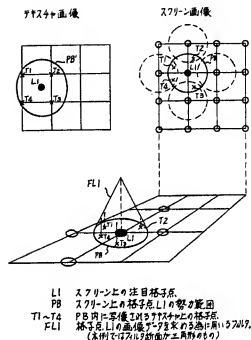


前手バス用プログラム・フロー

第 11 図

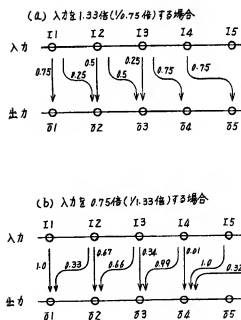
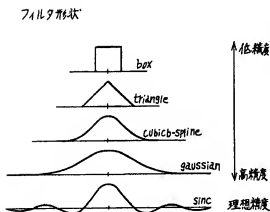


第 12 図

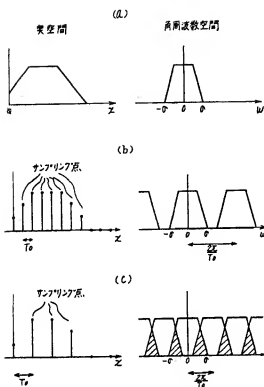


第 14 図

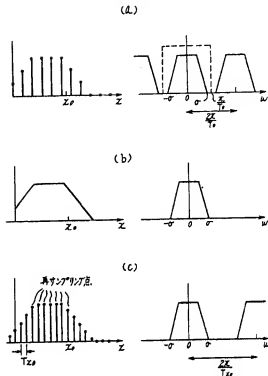
第 13 図



第 15 図



第 16 図



第 17 図

